

The Developer Group
Phlab – The Common Address Book

Phlab is just a very simple application that is being used as a vehicle to demonstrate the differences in developing an application in the following environments (builds):

Windows Application	Time (mins)	Presenter
1) Delphi – win 32 (e.g. D5)	55	A
2) BDS (Borland Developer Studio) - using Delphi ,VCL for .NET	15	A
3) BDS - using Delphi, FCL (not VCL for .NET)	55	B
4) BDS - using C#	45	C
5) Visual Studio 2005 - using C# (.NET 2.0)	30	C
Total:	3hrs 20mins	

Web Application	Time (mins)	Presenter
1) Delphi – win 32 (e.g. D5) - web broker	55	D
2) BDS - using Delphi, VCL for .net - (Web broker not ported)	0	
3) BDS - using Delphi, FCL (not VCL for .NET) - asp.net	55	E
4) BDS - using C# - asp.net	45	F
5) Visual Studio 2005 - using C# (.NET 2.0) - asp.net	30	F
Total:	3 hrs 5 mins	

Notes:

- 1) The Group Leader doing the initial introduction should:
 - a) Demonstrate a working/partial working version of the application (Phlab)
 - b) Show the database table schemas being used.

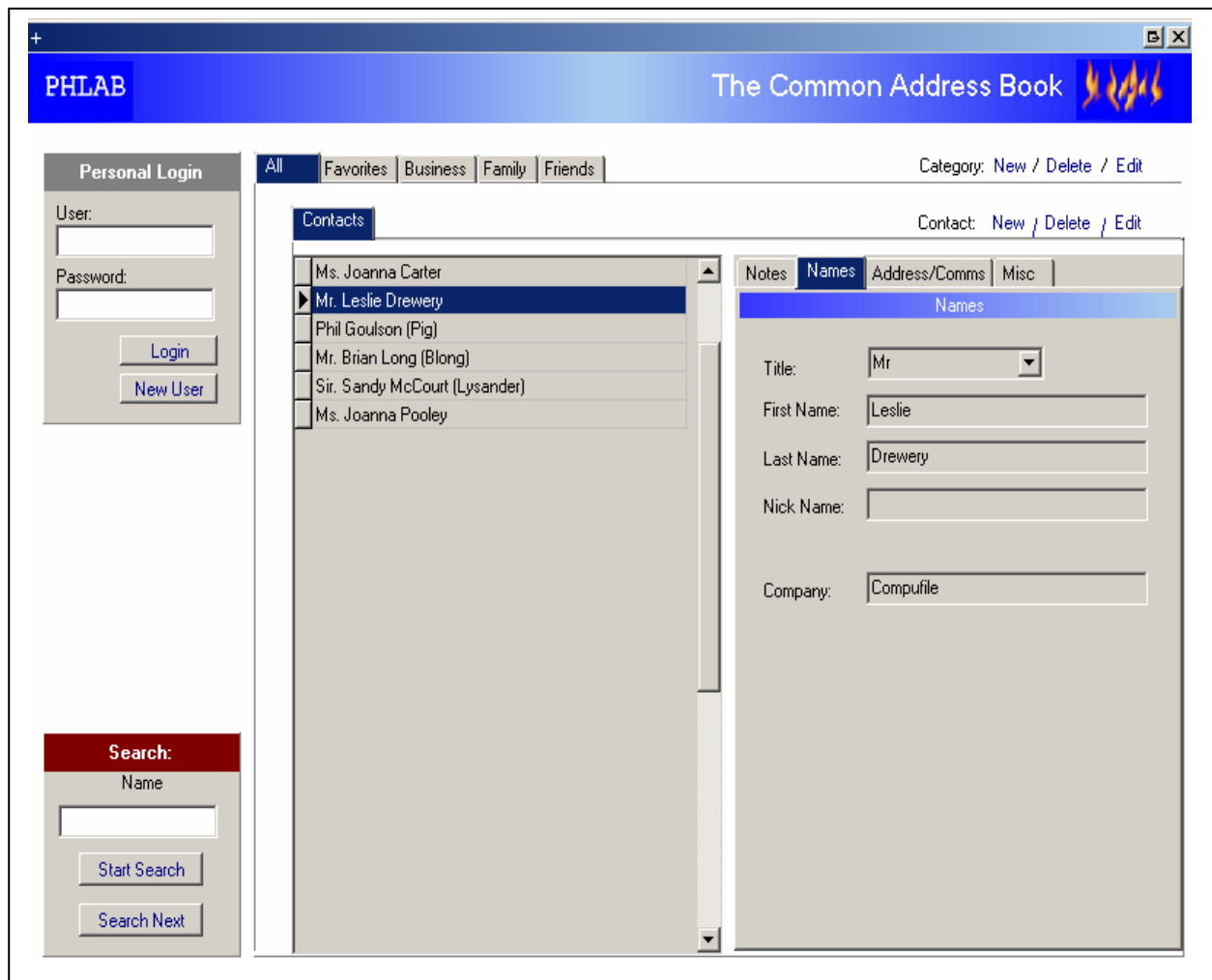
- 2) It really does not matter if every build uses a different database, however it is suggested that presenters use one of the following: MS SQL Server, Firebird, Interbase, MS Access.

- 3) In all builds it is important to show how an application/project is created. This is regardless of the similarities to other builds, it is important to show these similarities.

- 4) Phases
 - a) For all builds it is important to build the application in defined phases.
 - b) Each build will probably have different defined phases as determined by the presenter.
 - c) The presenter should highlight these phases at the start of their presentation.
 - d) It is not necessary for a presenter to totally complete each phase (on stage); after a presenter has demonstrated the basics and complexities of a phase, the presenter should produce code for a completed phase (created earlier) and then carry on to the next phase.

The Developer Group

Phlab – The Common Address Book



(Fig. 1)

1) Introduction:

Phlab is just a normal name/address system with a few quirks. It is not intended that Phlab should work in its entirety; it is just a vehicle for demonstration. But I should explain the basic operational concepts so that presenters can pick the functionality they want to implement.

I initially tried to design Phlab as if it were a ASP.NET application but threw that out of the window and wrote a demonstration version in D7.

The database I used was Firebird 1.5 using FreeIbComponents to access it. (I know it is old technology but I know it backwards).

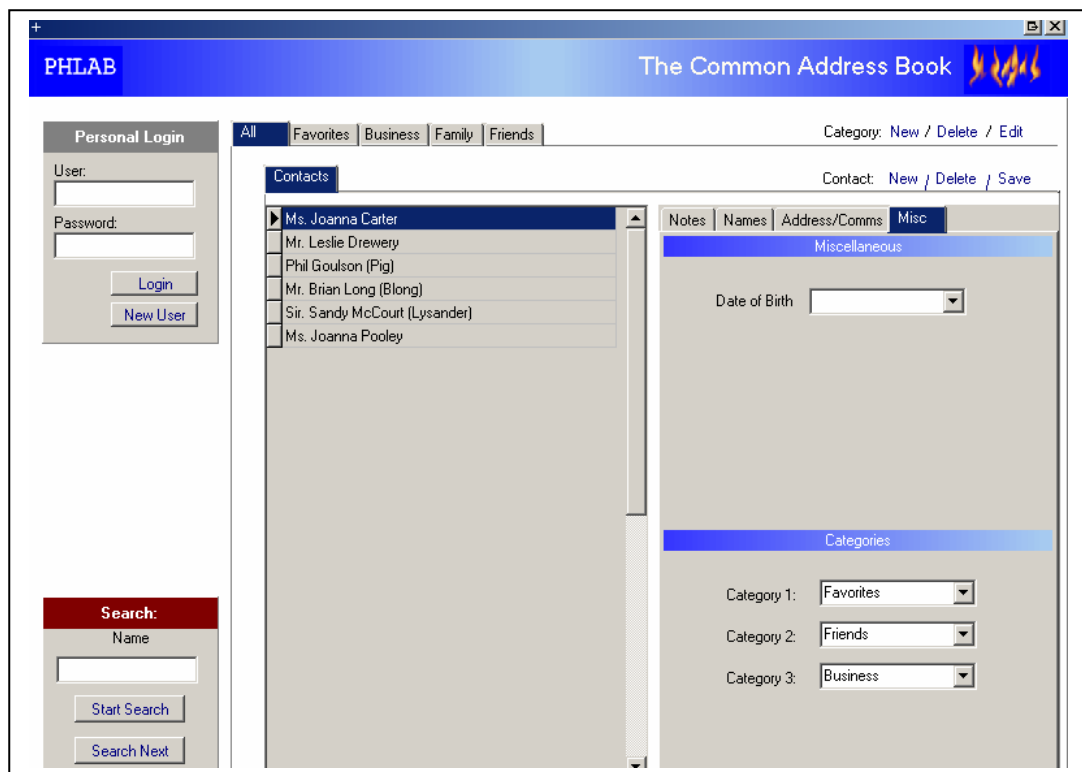
2) Login:

- a. The idea was, if a user did NOT login, they would be presented with a common address book, available to all. If a user did login then the address book would be personal to them.
- b. I never implemented the login/personal function in my demo. I treat everyone the same (common).

The Developer Group

Phlab – The Common Address Book

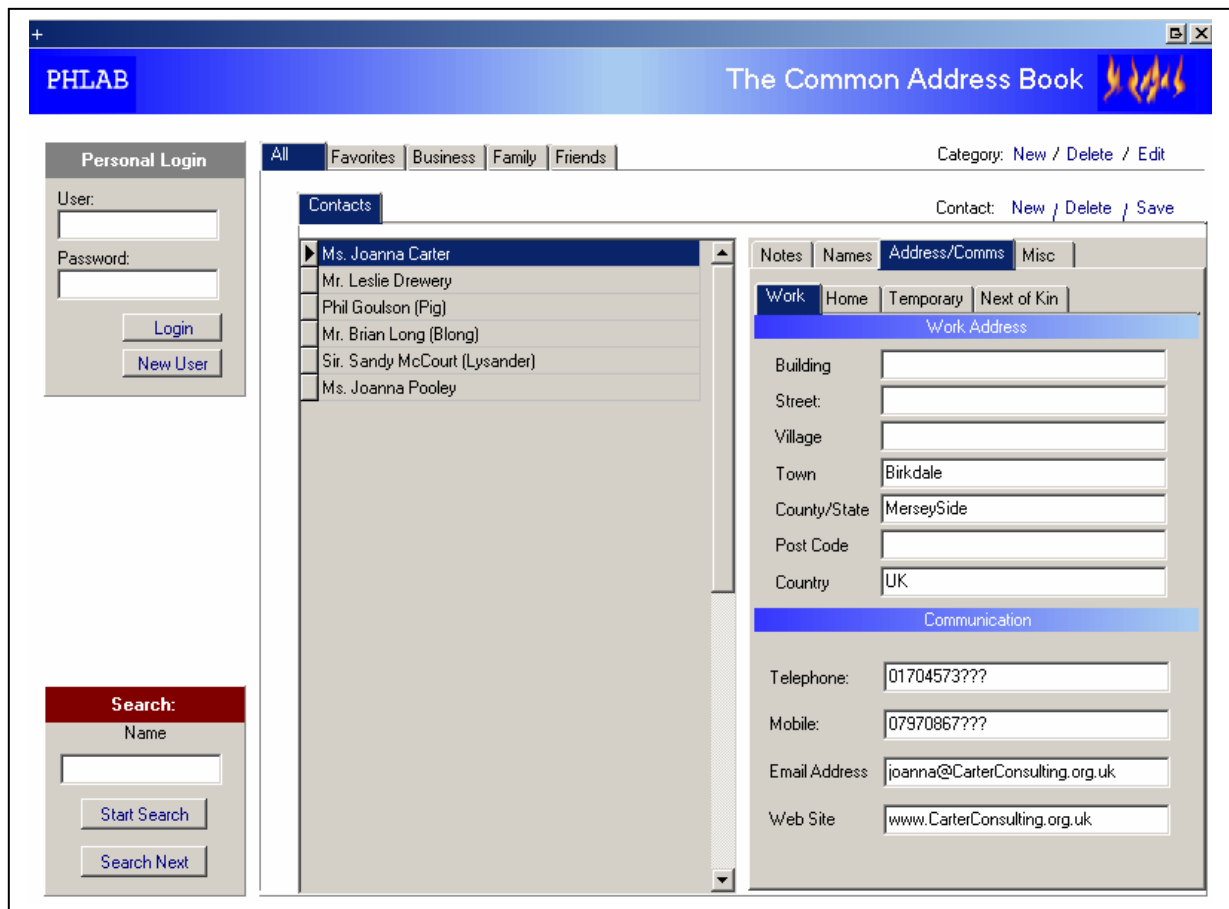
- 3) Categories:
- The Category Tab is loaded at runtime from the Category table. The DisplayNo Field is used to sequence the order in which the categories are displayed.
 - The concept is that as a user clicks a Category tab then the contacts associated with that Category are shown in the grid.
The ALL Category tab is obviously the exception; in this case all contacts are shown.
 - The Misc Tab of each contact (see fig 2) is where a contact is assigned the Categories to which it belongs.
 - The function to Add (new) / Delete / Edit categories has not been implemented in the demo. All that is required is to add a nice user interface to maintain the Category Table.



(Fig. 2)

The Developer Group

Phlab – The Common Address Book



(Fig.3)

4) Contacts:

- e. The Contacts table contains the basic fields required for a name/address application and is nothing special
- f. The only field of any relevance is the ContactNo field which is assigned a unique value for every Contact in the database.
- g. I have created a table called Control which contains the NextContactNo. I suggest you use this when generating a new contact.
- h. UserNo field: This should be assigned the value 0 for all Contacts that are a part of the Common Address book. i.e. Those contacts that do not belong to a specific user.
- i. The function to Add (new) / Delete / Edit contacts has been implemented in the demo and functions as follows:
 - i. When data for a contact is initially displayed, all the data fields are shown with the same background color (clBtface) to signify that they are not editable.
 - ii. When the New or Edit labels are clicked the data fields become editable and their background colour changes (clWindow) to signify this state.
 - iii. When the New or Edit labels are clicked the Edit label changes to the word “Save”.

The Developer Group
Phlab – The Common Address Book

- iv. When the Save label is clicked, contact data is saved and the Save label changes back to “Edit”.
- v. There is no Cancel New/Edit function – in demos you never do this.
- vi. Yes, you can press the New label without having previously pressed Save. This automatically saves the contact data and then creates a new contact.
- vii. When the New label is clicked the Title field in the Name tab in the Address/Comms tab is automatically selected.
- viii. The Delete label does what it says however Delete Confirmation is only requested if a Contact Name has been entered.

5) Contacts Address/Comms

- a. As can be seen in fig 2, this is another variable tab which is populated by the AddressTypes table, at runtime.
- b. As each AddressType is selected then the details relating to that AddressType are displayed.

6) Suggestions to the presenters

Because of time constraints it is important concentrate on the aim of this whole project and not on best practices. With this in mind I would suggest that you try to implement (Live coding) the following functionality and only expand on this if there is available time.

- a. Populating the Category tab with categories from the Category table.
- b. Edit a Contact so that their categories can be changed.

The Developer Group
Phlab – The Common Address Book

Table Structures

AddressTypes

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
PK	AddressTypeNo	SmallInt	
	AddressType	Varchar (30)	
	DisplayNo	smallint	used to sequence the order in which the AddressTypes are displayed

Category

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
PK	CategoryNo	SmallInt	
	Category	Varchar (30)	
	DisplayNo	SmallInt	used to sequence the order in which the categories are displayed

Contacts

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
PK	UserID	SmallInt	
PK	ContactNo	SmallInt	
	CompanyName	Varchar (40)	
	TitleNo	SmallInt	relates to the Titles table to get the Title field
	FirstName	Varchar (50)	
	LastName	Varchar (50)	
	NickName	Varchar (50)	
	DateOfBirth	TimeStamp	
	Notes	Text Blob	
	Categories	Varchar (20)	Contains a concatenation of the category fields (1,2 & 3). Used in the select query to determine if a Contact belong to a specific category Eg ,3,5, means that the Contact belongs to Category 3 and 5.
	Category1	SmallInt	
	Category2	SmallInt	
	Category3	SmallInt	

Control (one record)

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
	NextContactNo	SmallInt	

The Developer Group
Phlab – The Common Address Book

ContactDetails

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
PK	UserNo	SmallInt	
PK	ContactNo	Smallint	
PK	AddressTypeNo	smallint	Relates to AddressTypes table
	Building	Varchar (50)	
	Street	Varchar (50)	
	Village	Varchar (50)	
	Town	Varchar (50)	
	County	Varchar (50)	
	Country	Varchar (50)	
	PostCode	Varchar (30)	
	TelNo	Varchar (30)	
	Mobile	Varchar (30)	
	EmailAddress	Varchar (60)	
	Website	Varchar (60)	
	Fax	Varchar (30)	

Titles

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
PK	TitleNo	SmallInt	
	Title	Varchar (20)	
	DisplayNo	SmallInt	used to sequence the order in which the Titles are displayed

Users (shown for completeness but never used in the demo)

	<i>Field:</i>	<i>Type</i>	<i>Notes</i>
PK	UserNo	SmallInt	
	UserName	Varchar (30)	
	UserPassword	Varchar (30)	
	LastLoginDate	TimeStamp	

PK = Primary Key